DOCUMENT RESUME

ED 181 799 HE 012 191

AUTHOR Farmer, James; And Others
TITLE A Guide to the Design of Student Financial Aid Systems.
INSTITUTION Systems Research, Inc., Washington, D.C.
REPORT NO SP-79-0013
PUB DATE Jun 79
NOTE 50p.
AVAILABLE FROM Systems Research, Inc., 1030 15th Street, Suite 300, Washington, DC

EDRS PRICE MF01/PC02 Plus Postage.
DESCRIPTORS *Automation; Compliance (Legal); *Computer Programs; Computer Science; Cost Effectiveness; *Data Processing; Evaluation Methods; Flow Charts; Higher Education; *Models; On Line Systems; Recordkeeping; *Student Financial Aid; Systems Analysis; *Systems Development

ABSTRACT

Characteristics of student financial aid systems are reported and a framework for evaluating student financial aid systems and classifying major data processing designs is presented. The goals of a student financial aid system are discussed, including improved service to students, increased utilization of available funds, compliance with regulations and policies, and reduced costs. Reasons for automating student aid processing, such as to cope with workload increase, are examined and the basic functions of the system are identified. Areas of systems design--including methods of processing, basic systems designs, modes of data processing, and on-line applications--are surveyed. The relationship of student financial aid systems to other institutional systems and inter-system communication and database integration is examined. The file structure of the system, including alternative approaches for file storage is reviewed. Twenty-one charts illustrate various phases of the systems design. (PHR)

****
* Reproductions supplied by EDRS are the best that can be made *
* from the original document. *
****

ED181799

"PERMISSION TO REPRODUCE THIS MATERIAL HAS BEEN GRANTED BY

James Farmer

TO THE EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)."

U.S. DEPARTMENT OF HEALTH, EDUCATION & WELFARE
NATIONAL INSTITUTE OF EDUCATION

THIS DOCUMENT HAS BEEN REPRODUCED EXACTLY AS RECEIVED FROM THE PERSON OR ORGANIZATION ORIGINATING IT. POINTS OF VIEW OR OPINIONS STATED DO NOT NECESSARILY REPRESENT OFFICIAL NATIONAL INSTITUTE OF EDUCATION POSITION OR POLICY

HE 012 191

SYSTEMS RESEARCH INC.

SR

A GUIDE TO THE DESIGN OF STUDENT FINANCIAL AID SYSTEMS

James Farmer
William J. Collaro
Gary S. Gamso
Richard L. Tombaugh

SP 79-0013

June 1979

Systems Research Inc.
Washington, D.C. 20005

ACKNOWLEDGMENTS

The authors wish to express their appreciation to Don Holec, Director of Financial Aid, Purdue University; William Murphy, Director of Financial Aid, Glassboro State College; Tom Hartigan and Brett Lief of the New Jersey Department of Higher Education's Office of Student Assistance; and Steve Pritz, Associate Director of Financial Aid, University of Florida at Gainesville, who spent many hours in their computer centers explaining the intricacies of student financial aid and, perhaps even more important, providing an insight into the motivations and needs of the student financial aid professionals. This report records many of the observations made by these student financial aid leaders.

The students in the Management Information System course at Harvard University's Graduate School of Education have made a contribution by insisting that complex data processing issues could be made understandable to administrators and by enduring case studies and research projects demonstrating the impact of these information systems on the administration of a college or university.

This report is founded on many contributions to student financial aid administration, particularly the work of Francis Keppel and his colleagues on the National Task Force on Student Aid Problems, which have been made in the past decade. The authors are, however, responsible for the synthesis of these materials and judgments in this report.

PREFACE

Because of new and expanding student financial aid programs, the workload of student financial offices has increased significantly in the past ten years. There are additional regulations and policies on accountability which have made the task more complex and, perhaps, less personal. In most institutions, the student financial aid officer is responsible for 20 to 50 percent of the gross revenue of the institution, thus becoming a major implementor of institutional policy. Student fianncial aid officers are increasingly turning to automation to reduce the clerical burden so that their offices can continue to service students.

This report describes the characteristics of those systems, provides a framework for evaluating student financial aid systems, and classifying major data processing designs. These materials should assist student financial aid staff in accurately specifying their needs and in assessing data processing alternatives which are available. It may also be useful to information system specialists who must communicate the characteristics of their design to student financial aid staff and identify the design tradeoffs which have been made.

The report was prepared for the Annual Conference of the National Association of Student Financial Aid Administrators (NASFAA) held in Boston, Massachusetts, July 1979.

James Farmer is an Associate of the Harvard University Graduate School of Education; Gary S. Gamso and William J. Collard are with Systems Research Inc.; and Richard L. Tombaugh is Vice President, Educational Methods Inc.

Papers are reproduced as a courtesy for Systems Research Inc. staff and associates. The views expressed in this paper are not those of Systems Research Inc. or any of their clients; the authors are solely responsible for the content of the paper.

CONTENTS

ACKNOWLEDGEMENTS iii

PREFACE v

INTRODUCTION 1
- Issues 1
- Goals of A Student Financial Aid Systems 2
- Basic Functions 6

FUNCTIONAL ANALYSIS 11
- Methods of Processing 16
- Basic System Designs 19
- Modes of Data Processing 23
- On-Line Applications 27

RELATIONSHIP OF STUDENT FINANCIAL AID SYSTEMS TO OTHER INSTITUTIONAL SYSTEMS 31
- Inter-System Communication 31
- Database Integration 35

FILE STRUCTURE 39
- Issue 39
- Alternative Approaches 39

SUMMARY 45

INTRODUCTION

ISSUES

Each year there are new and expanded student financial aid programs which increase the workload on student financial aid offices at colleges and universities. Added regulations and broader policies of public accountability have also sharply increased the administrative workload. Applicant increases of 30 to 80 percent, program award increases of 50 to 100 percent, and additional compliance requirements of 20 to 50 percent per year have been typical for the past two to five years. Unfortunately, these workload increases have occurred when institutions with constrained budgets and limited personnel flexibility have been unable to respond with proportional staff increases.

Student financial aid has also become a major institutional financial consideration, as these funds typically account for 20 to 50 percent of the institution's gross revenue. Institutional student financial aid has become a major vehicle for implementing differential pricing, selective recruiting, and capacity utilization. While student financial aid staff have traditionally viewed themselves as assisting students to attend and remain in higher education, emphasizing the counseling aspect of their responsibilities, they have become implementors of broad institutional policies and, in some cases, are responsbile for the financial health of the institutions. Clearly, student financial aid has major new roles in the management of the institution.

With new responsibilities and additional workload, student financial aid offices have increasingly turned to automation to relieve the clerical burden, to improve service to students, and to increase the scope and timeliness of management information. Student financial aid staff must specify the requirements for data processing systems and evaluate alternatives, implement these systems with very limited resources, and develop new working relationships with the business office and top management of the institution. Those that have been unable to respond have found that either the business office or the computer center has initiated efforts to automate student financial aid, the state has expanded its administrative functions, or some of the basic and important objectives of student financial aid are not being met.

This guide is intended to assist the student financial aid staff in their efforts to improve both administration and management of student financial aid and to better respond to the expanded role appropriate to these professionals in the institution. A number of institutions have made large commitments of resources to develop student financial aid systems. Almost all have failed to meet their objectives or the expectations of the student financial aid staff. Similarly, a number of firms

have attempted to develop complete systems, yet none has incorporated all of the functions which have been identified by the student financial aid staff. These failures have not occurred either because of lack of commitment or lack of competence, but rather from a lack of understanding by the data processing developers of the magnitude and nature of student financial aid processing and the inability of the student financial aid professionals to articulate, at that time, the needs of the system in the detail and form that would be completely understood by the data processing professionals. These attempts have made major contributions to student financial aid systems. Many separate functions have been successfully developed and implemented. These provide the basis for today's new systems. The classification of functions, data on processing workloads, and processing effectiveness have all come as the result of developing expertise in the student financial aid community.

GOALS OF A STUDENT FINANCIAL AID SYSTEM

The goals of a student financial aid system are improved service to students, increased utilization of available funds, compliance with regulations and policies, and reduced costs. (See Chart 1.) Students have two basic needs for student financial aid service. First, they need advice on availability and eligibility, procedures for application for student financial aid and general information on the costs and methods of financing. To support this traditional counseling role, student financial aid staff need immediate access to student academic records, student financial records, financial aid transcripts, and loan records.

Second, the students need prompt and accurate administration of the student financial aid system both to provide them funds for their college education and to assure that they comply with the federal, state, and institutional requirements.

From the institutional prospective, the student financial aid system will improve the effective utilization of available funds. This is done in two ways. By being able to analyze the needs for student financial aid, to plan institutional policies based on information from the student financial aid system, and to closely monitor use of funds, the institutions are able to use all available funds and, should additional funds become available, to immediately deploy these funds to first priority students. Because student financial aid is required for some students to attend the institutions, and may be a significant incentive to others to attend the institution, effective utilization of student financial aid funds frequently increases enrollment at the institution, particularly those students to whom the institution gives preference.

As the number of student financial aid programs has increased in the past few years, so have the requirements to comply with all the regulations. Since the award amount of many student financial aid programs is in turn dependent upon the amount of some other student financial aid program, and because of the broad requirements for satisfactory progress,

GOALS FOR A STUDENT FINANCIAL AID SYSTEM

- IMPROVED SERVICE TO STUDENTS
- INCREASED UTILIZATION OF AVAILABLE FUNDS
- COMPLIANCE WITH REGULATIONS AND POLICIES
- REDUCED COSTS

CHART 1

assuring compliance has become a difficult task and must be tested frequently since any change in the student's enrollment financial condition, or other resources is likely to impact the award amount of a number of different programs. To assist in this complex and tedious task, automated compliance monitoring has become important.

At the same time there has been a sharp increase in the number of applicants for student financial aid and the number of awards the average student financial aid recipient receives and in the compliance requirements, institutions have come under increasing pressure to reduce the cost of administration. Since student financial aid developed after the period of affluence in colleges and universities, most financial aid offices have only spartan staffing. This staff has had to absorb increasing workloads each year and is being asked to continue to do so even though the workload continues to mount much faster than in any other part of the institution. In order to achieve the goals of student financial aid and at the same time meet institutional constraints on fund availability, it is important to minimize the cost of student financial aid processing. This has become one of the primary goals of any student financial aid system. From a practical point of view, student financial aid offices have continued to devote more and more resources to the administration of student financial aid rather than as originally intended to assist students in entering colleges and universities and remaining there.

As shown on Chart 2, there are six reasons for automating student financial aid processing. First, most students and their parents have a higher expectation for service than in the past. They expect the financial aid office to respond immediately to any questions about the status of the student's awards or payments. Similarly, they expect by telephone or letter to determine whether all information has been provided and also the status of the awarding process. Since the first institutional contact for many parents and students will be the student financial aid office, it is important this contact be responsive and informative.

Student financial aid processing systems should also assist the financial aid office in coordinating with the institution's policies. As institutions focus their recruiting on specific types of prospective students, and as they pursue retention policies, they expect student financial aid to contribute to these institutional goals and to conform with institutional policies. This is difficult to do unless the financial aid office can provide information on financial aid applicants and the status of their funds, and also to make their awards strictly in accordance with these policies. While close coordination between admissions and student financial aid, and counseling and student financial aid has been a practice in private institutions for many years, increased coordination is also important for public institutions. A student financial aid system provides the data base and the methods for determining institutional policies, and through automated packaging, a method for implementing the complex policy for thousands of student applicants.

REASONS FOR AUTOMATING STUDENT AID PROCESSING

- STUDENT/PARENT EXPECTATIONS FOR SERVICE
- COORDINATION WITH INSTITUTIONAL POLICIES
- WORKLOAD INCREASE
- REPORTING REQUIREMENTS
- COMPLIANCE
- PLANNING

CHART 2

One of the advantages of an automated student financial aid system is the ability to absorb increasing workload without increasing staffing proportionately. As the number of applications increases, the institution has to depend upon the national processors to provide the basic information for students, automated interfaces with admissions and registrations systems to monitor status, and the capability to process large volumes of awards and communications with students without additional staff. Although at some point there clearly must be staff increases, these do not come as often or as large as would be necessary for manual processing.

The reporting requirements from federal and state agencies continue to mount and are likely to continue to increase as public attention is focused on student financial aid. The only way to achieve these reporting requirements without massive manual statistical analysis is to use the data base created by an automated system. Most of the reports can now be completed by student financial aid systems without having to resort to manual data entry or manual tabulations.

Since compliance essentially requires the constant monitoring and recomputation of student financial aid, it represents a major workload for the institutuion. The automation of compliance should permit an institution to have confidence in its student's financial awards and demonstrate a high level of compliance with regulations and policies.

As institutions review policies on admissions and student retention and implement more financial planning, the student financial aid data base becomes one of the most important and useful data bases for planning for the future. Since student financial aid constitutes 20 to 60 percent of the total revenue of the institution, even in public institutions, it is clear that student financial aid is one of the most important factors in institutional health.

Basic Functions

In the course of developing comprehensive student financial aid systems, a classification of the basic functions has developed. The general classification shown in Chart 3 is now used in the literature by several of the companies developing proprietary systems. Nine basic functions are typically represented in software.

Application tracking is significantly different from its admission system counterpart. A completed student financial aid application includes a number of documents, such as need analysis, copy of parents' income tax form, and application for the federal BEOG program. Tracking is essentially monitoring an inventory of received documents. It may also include an interface with the admissions system to include "acceptance" as a required activity. The process is further complicated by the fact that different categories of applicants may be required to provide

BASIC FUNCTIONS

- APPLICATION TRACKING
- FINANCIAL ANALYSIS
- EVALUATION
- PACKAGING
- NOTIFICATION
- DISBURSEMENT
- COMPLIANCE
- FUNDS MANAGEMENT
- REPORTING

CHART 3

differing combinations of documents or actions to be considered "complete."

Some sort of financial analysis is required to support the award of financial aid on the basis of need. This is often called need analysis. For most students three types of analysis are required. First, the federal government will develop an index for the award of the Basic Educational Opportunity Grant (BEOG) program. Since this computation frequently comes after the student has received his other awards, most national processing services provide, and most institutions need, an estimated Basic Education Opportunity Grant Index. Second, the so-called Uniform Methodology, with its many institutional options, is used by the national processors as a method of determining need. This U.S. Office of Education-approved process is also incorporated in proprietary packages to provide the capability for institutions to either compute this value from institutional data or perform a recomputation using data from a national processor. Third, a special state index is required for several of the state programs. As a condition of eligibility for many state programs, a financial analysis different from both the Uniform Methodology and the federal BEOG computation must be performed. Since each of these methods of determining need is likely to change each year, the institution is faced with maintaining three separate and changing methods of determining need. The financial analysis function is usually isolated in the system design since U.S. Office of Education approval is required for any software which performs a campus-based need analysis.

A second need-related function is evaluation, which relates the financial analysis to the student budget. The process of determining a student budget is becoming increasingly complex as each component of the student budget is carefully analyzed. This analysis and development of the student budget and a comparison with the family contribution leads to an evaluation of the student's need for financial aid. Often this need is communicated directly to the student in the form of a preliminary evaluation which establishes general eligibility and the maximum level of financial aid for which a student would be eligible.

Because most students receive not a single award of student financial aid but a number of awards from various state, federal, institutional and private sources, the combination of awards is generally called a "package." Each of the awards has criteria for eligibility. Some programs are entitlements, in the sense that the student will receive the award regardless of the institutional policy. On the other hand, many awards are based on institutional priorities, admissions and retention policies, and availability of funds. These considerations must be taken into account in determining a student's package. There are now several packaging programs which permit an institution to implement these constraints and policies. As packaging programs have been used, a need has been identified to modify student packages in accordance with the availability of funds, for changes in regulations--often regulations are changed after awards are made--and for changes in student circumstances.

The major workload for the packaging program is no longer initial packaging of the student, but revision of these packages in accordance with the changing circumstances. Since repackaging often occurs two or three times during an academic year affecting different students, this software tool needs to be particularly flexible to meet the continuing needs of the student financial aid office.

Although there are conflicting points of view by student financial aid offices on notification, students must receive some notice of student financial aid. In some institutions this notice of award is given at the same time admissions response are made or provided immediately to continuing students upon the indication of their return to the institution. Other institutions elect to notify students after some reasonable assurance that the funds are available for these aid programs. Notifications may contain detailed information on the student's financial circumstances as evidenced by reported data and analysis, a detail or summary student budget, a list of awards given to the student outside of the institution, and the awards made by the institution, or they may contain only awards. There are several different forms and times for notification; all must be supported by a notification program so the student financial aid officer will have some flexibility in his use of this capability.

Most student financial aid offices do not make disbursement of funds to students. It is usually preferable for an institutional business office to handle all disbursements of funds or crediting of student aid funds to students' accounts. An automated interface between the student aid system and student accounts receivable reduces the workload of manual data entry. The disbursement function in student financial aid primarily consists of scheduling payments to the student and monitoring continued eligibility. On the other hand, it is necessary, where business office systems are not yet fully developed, to be able to provide checks or vouchers to students. The disbursements function should include the abilities to schedule payments, to make payments if the business office system is not available, and to store information on payments which have been made so that the financial aid staff can respond immediately to student inquiries.

The federal government has always audited federal student aid funds and now requires institutions to have an outside audit. Most state student aid programs have audit staff. Federal and state regulations mandate institutional review of student applications and "due diligence" in collections. Notwithstanding press reports, almost no institutions have been penalized as a result of these audits. But both the institutions and students are under the threat of audit results. Claims for federal or state guarantees on loans can be denied, institutions may have to repay funds awarded in error, or students may have to refund previously received student financial aid. To prevent these results, many institutions have begun pre-payment and post-payment reviews, and comparisons of student financial data from several sources. These compliance activities should be included in a student financial aid system.

Financial aid and business officers must monitor the status of student aid funds carefully. Sufficient "awards" must be made to fully utilize the available funds and assist as many students as possible; on the other hand, there must not be so many awards that insufficient funds will be available for payment. The funds management function typically reports on the status of funds, giving the amounts which have been awarded, accepted by students, committed to pay by institutions, and paid. These latter two categories correspond to the accountant's encumbrance and payment (liquidation of the encumbrance). This analysis must be done by both student financial aid program and institutional accounts.

Reports on student financial aid activity are needed by many institutional offices--business, admissions, budget, and planning offices, as well as the financial aid office. Reports are required by external agencies--federal and state agencies--and lenders. The scheduled and formal reports are usually produced by a "reporting" module of programs, supplemented by a report writer or query language.

This classification of basic functions is useful for analyzing student financial aid systems and structuring the discussion of financial aid processing. Other classifications are possible, but this schema appears to be the most common at this time.

SYSTEMS DESIGNS

FUNCTIONAL ANALYSIS

In a complex system like one necessary for the administration of student financial aid, the basic functions are often represented by groups of computer programs, called "modules." These modules can be designed like building blocks so that they fit together to form a complete system. Or, for processing efficiency, some of the functions may be combined together.

But for purposes of design, it is most important to identify those functions which are to be done by the system. Software can perform three tasks. First, the software can perform a computation. This may be a test for application completion which merely requires that the appropriate documents be selected for the student and this list compared to the list of documents received. (Activities such as "acceptance" may be included.) Or it may be the computation of parental or student contribution from financial data. Both are computations in the sense that a set of rules, or an algorithm, is followed and the results derived from available or assumed data.

Second, the software can prepare letters, labels, or telephone call lists which result in a communication with the student or his parents. These can be routine forms or very specialized letters relating to the specific individual and his circumstances. The telephone call list can be general or it can identify the substance of the call.

Third, the software can produce reports, either for the financial aid office, other institutional users, or external agencies.

In Charts 4 to 7, each one of the basic functions is analyzed to list the typical computations performed by a comprehensive system, the communications with students or their parents, and reports for the financial aid office, other institutional offices, and external agencies such as the Department of Health, Education, and Welfare; Office of Civil Rights; the U.S. Office of Education; state agencies; and lenders.

The list of tasks and functions can be used to develop specifications for institutional systems or a comparison of financial aid systems. Not all "output" or products--communications or reports--will be required by each institution. Nor must all computations be performed by the software. Such computations as the Basic Educational Opportunity Grant Index can be obtained from the Student Eligibility Report (SER) provided by the federal government, a magnetic tape or listing from the federal processor, the national need analysis services, or hand computations. The appropriate

FUNCTIONAL ANALYSIS

COMPUTATION	COMMUNICATIONS	REPORTS
APPLICATION TRACKING		
COMPLETION	ACKNOWLEDGMENT OF DOCUMENTS	APPLICATION STATUS
	REQUEST FOR DOCUMENTS OR ACTION	FILE LABELS, LISTS
FINANCIAL ANALYSIS		
STUDENT/PARENT AND FAMILY CONTRIBUTION	REQUEST FOR ADDITIONAL INFORMATION	FINANCIAL STATUS
BEOG INDEX	FINANCIAL ANALYSIS	FINANCIAL PROFILES
STATE INDEX		

CHART 4

FUNCTIONAL ANALYSIS

COMPUTATION	COMMUNICATIONS	REPORTS
EVALUATION		
STUDENT BUDGET	INELIGIBLE NOTICES	EVALUATION STATUS
BEOG BUDGET	NOTICES OF ELIGIBILITY	
GROSS NEED		
OUTSIDE RESOURCES		
ELIGIBILITY (NET NEED)		
PACKAGING		
PRIORITY	PACKAGING WORK SHEETS	PRIORITY LISTS
GROSS NEED	FINANCIAL AID TRANSCRIPTS	PROCESSING LOG
NET NEED		PACKAGING PROFILES
UNMET NEED		
AWARD PACKAGE AND REPACKAGING		
PAYMENT DISTRIBUTION OFFERS		
NOTIFICATION		
GROSS NEED	AWARD NOTICES	NOTIFICATION STATUS
NET NEED		
AMOUNT AWARDED		

CHART 5

FUNCTIONAL ANALYSIS

COMPUTATION	COMMUNICATIONS	REPORTS
DISBURSEMENTS		
PAYMENT SCHEDULING AND AND RESCHEDULING	STUDENT ACCOUNT STATEMENT [CHECK OR VOUCHER] *	PAYMENT SCHEDULE JOURNAL
DISBURSEMENT ELIGIBILITY		FUNDS SUMMARY
AID PROGRAM TOTALS		REGISTER
FUNDS MANAGEMENT		
FUND BALANCES		FUND STATUS BY: FUND ACCOUNT CAMPUS PROGRAM
FUND UTILIZATION RATES		FUND UTILIZATION ACCOUNT RECONCILIATION
COMPLIANCE		
AWARD ELIGIBILITY		EXCEPTION LISTS
BEOG AWARDS		
STATE AWARDS		
COMPLIANCE EXCEPTIONS		
POLICY EXCEPTIONS		
IRS 1040 DATA COMPARISONS		
FEDERAL BEOG DATA COMPARISONS		
DOCUMENT VERIFICATION		

CHART 6

*Recommended output of the business office system, rather than financial aid system

FUNCTIONAL ANALYSIS

COMPUTATION	COMMUNICATIONS	REPORTS
REPORTS		
SUMMARY TOTALS		USOE FISCAPP REPORT
STATISTICS		DHEW OCR REPORT
TABULATIONS		STUDENT PROFILES
		FUND USE REPORTS
QUERY LANGUAGE		
SUMMARY TOTALS	FILE CONFIRMATION	FORMATTED LISTS
STATISTICS		LABELS
		GENERAL REPORTS
		RECORD DISPLAY

CHART 7

source depends upon the use of the computation, when in the processing cycle it will be required, the cost of alternate sources, and documentation which may be required for subsequent reviews and audits.

The list of tasks and functions can also be used as the basis for analysis to determine which functions would be automated. Depending upon the type of students, the institutional policies, and the workload, there will be priorities for development and implementation. Because of the high workload, complex processes, and staff workload, implementation is typically done in phases over a long period of time.

Since a query language is typically not part of the reporting module, but a separate software package associated with a database management system, it is listed separately in Chart 7. If a query language is not available, many of these routine reports must be separately provided.

New functions will be added to the list. The U.S. Office of Education is establishing procedures for confirming attendance through an exchange of tapes in accordance with prescribed formats and procedures. State guarantee agencies require status reports from institutions and provide data to lenders. This exchange of data will be added to student financial aid systems in the near future either as additional tasks in current functions or as new functions. The choice will depend upon the procedures which are developed.

METHODS OF PROCESSING

There are three basic methods of processing student financial aid listed in Chart 8. Most frequently, student aid processing is described in the "traditional" step-by-step process historically associated with the private liberal arts institution. The process is described as receiving all applications by a certain date, obtaining a need analysis for all applicants, determining eligibility and packaging at one time for all applications, sending award letters on one date, and disbursing aid through the business office from a list of awards provided by student financial aid. While this is the most frequent explanation, the step-by-step process is almost never seen at any institution, public or private. Rather, packages are continually modified due to changes in circumstances, student aid awards are made several times to match admissions actions and special offers to students wanted by the institution, recomputation of need is done after professional review, and award types and amounts are changed in consonance with fund availability. In addition, processing must be coordinated with federal and state processing, multiple or continuous admissions processing, and staff availability.

Perhaps one of the most confusing aspects of student financial aid has been using the "step-by-step" approach in explaining processing when the data processor finds reality much more complex. The "step-by-step" process is, from a data processing standpoint, relatively simple. The

METHODS OF STUDENT AID PROCESSING

- "TRADITIONAL" STEP-BY-STEP
- CYCLIC
- CONTINUOUS

CHART 8

process is never repeated, the student does not "recycle" through any of the steps, and there are no problems of data currency. Yet practice violates all of these assumptions.

The most common processing method is the cyclic method. A complete financial aid cycle is processed frequently, often each week. A student will begin processing from the last point and, if data are now available, the processing will continue. Thus a student who was in application tracking would, if the last document for completion were received, be evaluated and if eligible be packaged. The process for that student may terminate for this cycle pending a professional review or may be held pending availability of funds. When these are resolved, for this student, processing will continue at the next cycle.

While conceptually, cyclic processing may appear like the "step-by-step" processing, there are significant differences. First, control information must be maintained to identify current status. Second, the scheduling logic for processing must be included in the software instead of being determined external to the system. Third, procedures must be available to "recycle" the student from any point at any time.

Theoretically, a student making a complete application would have an award notice at the conclusion of the first cycle of processing. Rarely does this happen. Data are frequently missing for one or more steps of the processing cycle, professional review may be required, or special handling may be required. Also, good system design permits the student aid professional to override any of the steps: to make a student eligible who had not yet received citizenship, to make a special award to a student who otherwise would not be eligible, or to adjust financial need. The system must be able to stop processing the student at the appropriate step, accept the judgments of the student aid staff, and continue processing without the system "overriding" the judgments of the staff.

While conceptually cyclic processing is simple, the vagaries of student financial aid processing make it difficult in practice. In evaluating a system design, it is best to take "special" cases through the process to see if the software can perform the required functions. The most frequent failures are the inability to halt processing pending a professional review, the lack of support for manual processing at specific steps, and incomplete status information in terms which financial aid staff can interpret.

The third method of processing is continuous. This might be called an "instantaneous" cycle. Each individual student is processed from the current step through the last step for which data are available. This approach is used for "on-line" processing where the results are needed immediately. For example, a change is made in the student financial data and a revised financial analysis, preliminary eligibility notice, and award letter are produced immediately. These are the results of the

need analysis, evaluation, packaging, and notification functions or steps in the processing cycle.

In practice, it is difficult to design a continuous student financial aid system for three reasons. First, student aid is sufficiently complex that this processing requires large computer main memory, high input-output for the extensive tables, and compared to other business applications, extensive central processor time. Unfortunately, these are the demands which cause momentary overload even for large-scale computer systems. This may result in sharply increased response time for other users. As the price of computers continues to decrease, the high capacity required for student aid will become more cost-effective. (To understand the impact on a computer, consider the difference between changing the address on a driver's license--a frequent transaction--and the thousands of steps necessary to perform a financial analysis, evaluation, and packaging. This is further complicated by federal and state authorities using tables which are easy for humans instead of formulas which are easy for computers. The effects of a single data element change in student financial aid can be the computer equivalent of ten thousand simple transactions.)

The principal difference between cyclic processing and continuous processing is the organization of the processing steps. The cyclic method takes all students through each function, one step at a time. The continuous method takes one student through the process, one student at a time.

BASIC SYSTEM DESIGNS

There are many different student aid systems on the market with wide price variations. The primary difference in these systems is the basic system design (see Chart 9). The most primitive system, typically found in designs for minicomputers, is data entry and reporting. Literally all data elements are entered and reports are prepared by sorting and summarizing these data elements. All computations are done external to the system, and scheduling of processing is done by the user.

Though primitive, these systems can be cost-effective. Because of the amount of reporting which must be done, it frequently becomes cost-effective merely to enter the data for the resulting reports required for external agencies. A well-designed comprehensive system can be operated in this mode. This mode may be the first step in implementation of a more sophisticated system. The user should be aware of the difference between operating a primitive system, where data entry and reporting are the only capability, as compared to a comprehensive system which is operated in this mode, but has yet unused capability.

A more sophisticated system will be able to perform computations. The determination of general eligibility of the student is an example of a simple computation. Need analysis is a more complex computation.

BASIC SYSTEM DESIGNS

- DATA ENTRY AND REPORTING
- DATA ENTRY, COMPUTATION, AND REPORTING
- DATA ENTRY, COMPUTATION, PROCESS SCHEDULING, AND REPORTING

CHART 9

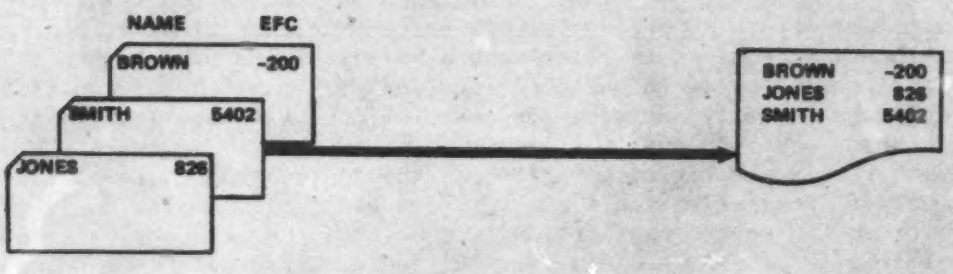
Some, or all, of the computations can be incorporated into the software and performed either by request of the user or by manipulation of "status flags" or dates.

A comprehensive system will include process scheduling. The software will determine what processing step is required and when it is required for each student. Thus, when an application is "complete," the software will process an evaluation for that student, and so on. Process scheduling is relatively straightforward except when a student's application or award must be "recycled" or when some of the steps are to be done manually, or some steps are skipped. Typically, the schedule depends upon the availability of data. When all documents are available and required activities performed, the application is complete. When all of the required data elements are available, a need analysis is performed. When the application is complete, the need analysis done, and the master file "complete," evaluation is performed. And so on. Processing logic can be difficult to define. Furthermore, the process scheduling will change throughout the year so that the software must be able to accommodate to these changes. Sometimes specific periods of time, such as the time between two letters requesting information, are changed depending upon the time of the academic year.

In developing a system design or evaluating a system, the most important area of a comprehensive system is the process scheduling. While data elements may be changed through table entries, while report formats can be changed either through table changes or simple program changes, and while definitions can be changed through control record changes, the process scheduling logic is embedded in the system design and only the most sophisticated designs will provide the capability to change the process during processing--omitting steps, adding or changing the amount of professional review, changing the packaging modes (as contrasted to policies), and changing the schedule for communications with the student. The financial aid staff should focus on this processing logic to assure that the institution financial aid policies can be implemented through the student aid system rather than the system dictating institution policies by its limited flexibility for process scheduling.

The three basic systems designed are illustrated in Chart 10. In the first case, the name and expected family contribution (EFC) are both data input and report output. In the second case, the expected family contribution is computed from financial data which are entered into the system. In the third case, financial analysis reports are scheduled and produced by the system for the two students which have complete data and an additional information request is generated for the student having missing asset data. The report is identical in all three cases and if the system is judged only for report output, the three systems producing the report would be judged comparable. Yet the second system would have the ability to recompute the expected family contribution for changed financial data, and the third system provides communication with the student that is important to relieve clerical burden.

DATA ENTRY REPORT



DATA ENTRY COMPUTATION REPORT

DATA ENTRY COMPUTATION PROCESS SCHEDULE REPORT

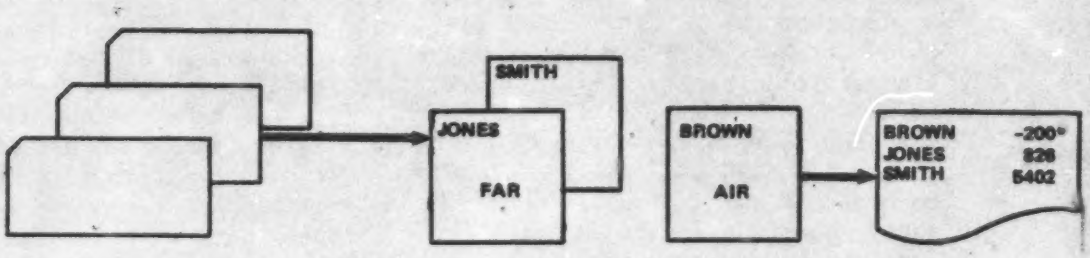


CHART 10

MODES OF DATA PROCESSING

Data processors design applications using three basic modes of processing. These are batch processing, timesharing, and transaction processing (see Chart 11). The three modes represent current software and hardware technology. Timesharing and transaction processing are both "on-line" in the sense that a user has either a printing terminal or a cathode ray tube (CRT or video) terminal. These terminals have keyboards and the software is designed to communicate in a dialogue with the user.

In batch processing, data are "keyed" into cards using a keypunch or onto some magnetic media such as discs or tapes. The processing is done from these data and control information often called "job control language" or JCL. This is the traditional mode of processing and still has several advantages. First, the software investment is usually less than for on-line systems. Second, the application system can be processed during non-prime (evening, night, or weekend) hours which is less expensive than prime time, or provides a balance to daytime use. Third, the system can accommodate large changes in workload without significant increases in cost or data entry congestion. Fourth, the system is less vulnerable to interruptions in processing than on-line systems. No complicated recovery procedures are required; the application system can be restarted from the last processing step. Fifth, these systems are simpler to use than on-line systems.

In on-line processing, keyboard terminals with either printers or video screens are located in user offices. This mode of processing is becoming more cost-effective and more widely used. There are several advantages. First, data entry costs are reduced since the user does not code data onto forms for subsequent keypunching. Second, since data entry is done by a user, the error rate is reduced from keypunching. While users may be less experienced in data processing and may make some additional keying errors, the user has sufficient knowledge of the application to identify more subtle errors. In addition, the user has the source document and can review the form if the data fail an edit or suggest an error. Thus data errors are resolved immediately. Third, there is immediate user access to data in files. This prevents clerical delay in accessing files for telephone calls or correspondence. Fourth, data are always timely since there is no delay until a "batch" of transactions is accumulated. There are some disadvantages. Computer capacity must be larger in order to accommodate peak periods during the prime hours rather than the average load over all available time. Database integrity and security become difficult with a large number of terminal users.

An institution with a sophisticated computer system may elect to implement batch processing student aid systems to permit easier implementation or to balance the computer workload by using less valuable computer capabilities.

MODES OF DATA PROCESSING

- BATCH
- TIMESHARING
- TRANSACTION PROCESSING

CHART 11

The two modes of on-line processing differ in their interaction with the user, the types of computers used, and the costs. Both are "interactive" in the sense that the application system communicates with the user. Timesharing communicates using one line at a time. That is, the application system sends one or more lines to the user and the user responds with one or more lines. Typically the lines "scroll" up the face of the user terminal. Transaction processing usually displays a form and indicates where the data should be entered. The form may also show available data so that the user can change the data. The content of a screen is typically called a "display."

Historically, timesharing has been available for scientific computers and is available for almost all minicomputers. The timesharing software is a component of the operating system and the application system is programmed similar to the batch system. However, the application system must contain the dialogue with the user. The two types are illustrated in Chart 12. With timesharing the name of the data elements is given on one line, the data are entered on the second, and so on. With transaction processing, the user enters a transaction number, and the screen is displayed with data element name or descriptions and places to input the data.

Typically used timesharing systems include IBM's TSO and CMS; Hewlett-Packard's MPE; Harris' Vulcan; DEC's TOPS10, TOPS20, and RSTS/E; and Prime's PRIMOS. The most widely used transaction processor in higher education is IBM's CICS. While historically, transaction processing was done on large systems, most minicomputer manufacturers are developing transaction processing software. Although IBM's Time Sharing Option (TSO) and similar timesharing systems on large-scale computers have been considered "inefficient," their efficiency has improved and, with decreasing computer costs, have become cost-effective for some applications. At the same time, transaction processors are becoming cost-effective on minicomputers. In general, timesharing applications are less expensive and more transportable between computers of different vendors. Transaction processor systems are more expensive and are not transportable between computers. Transaction processors are very cost-effective for data entry; timesharing systems require less maintenance and less software investment.

The choice of processing mode is likely not made because of student financial aid, but rather because of the institution's data processing system. If a transaction processing system is already installed, then a transaction processing version of the student aid system should be installed. If a minicomputer is purchased only for administration or only for student aid, then timesharing software would be more cost-effective and less likely to be obsolete. Transaction processing terminals are typically significantly more expensive, but have a more efficient communications technique. Timesharing terminals are inexpensive--about the same as an electric typewriter, but use a less efficient and sometimes slower communications technique.

BATCH

123456789 DOE/JOHN 1243 MAIN

TIMESHARING

SID	NAME		
123456789	DOE/JOHN		
ADDRESS	CITY	STATE	ZIP
"1234 MAIN"	HOMETOWN	OK	73044
PHONE			
4052820735			

"ON-LINE"

TRANSACTION PROCESSING

ID #: 123456789 NAME: DOE/JOHN

ADDR: 1234 MAIN CITY: HOMETOWN

ST: OK ZIP: 73044 PHONE: 4052821298

CHART 12

ON-LINE APPLICATIONS

There are three data processing functions which are performed on-line from a terminal. These are data entry and transaction processing, record display, and database query. In general, data are entered through "screens." Only data which may be changed are displayed. As the data are entered, edit messages are provided back to the user. These messages may be simple checks of the input data, such as verifying a number is numeric, data are longer than the field, or codes are valid, or more. In addition, the transaction processor can post new computed values to the file. For example, the change of a financial data element could cause a recomputation of the expected family contribution (EFC). This value could be added to the file. If this value is not retained but reprocessed later, this is called "memo posting." This design is most frequently used when the data are reprocessed later. In the banking industry, for example, the verification of a check may cause the account balance to be reduced through memo posting, but the account balance is actually changed only when the check itself is processed. Memo posting has three advantages. First, since values are subsequently recomputed, recovery procedures for a computer failure are simple and the on-line computations are less extensive. Second, it is not necessary to reconcile between memo balances and final balances. Other transactions could be entered from another source. Third, the processing is efficient since records do not have to be made for recovery. If all of the processing is completed with the transaction and never repeated, then this can be described as data entry and transaction processing. (Unfortunately, transaction processing is used in two senses, one to describe the data entry mode and one to describe how the data are used.) With this mode of processing, no reprocessing is done. Full capabilities for reconciling balances and restoring for failures are required; these may require extensive software and more computer power to maintain response time.

In addition to data entry, it is possible to retrieve the records of a student by having several "display screens" which provide data element names or descriptions and the data values. Thus screens are functionally equivalent to "pulling the files" on a student. This type of processing is particularly important when responding to telephone calls, where only a few seconds are available, or answering correspondence, where clerical efficiency is seriously eroded by delays in finding data.

Another useful function, but less time dependent, is a query to the database. For example, a query could list all students who have awards, but have not yet been paid. Or it may be a count of students with a particular award and status. Usually the database queries are done for ad hoc reports or a needed statistic. While comparatively inefficient, the results are available in a few minutes and do not require programmer intervention. Usually this type of report is done on a printing terminal or printed on the high speed printer at the computer center.

MODES OF ON-LINE PROCESSING

- TRANSACTION PROCESSING
  - DATA ENTRY, EDIT ONLY
  - DATA ENTRY, MEMO POSTING
  - DATA ENTRY, TRANSACTION PROCESSING
- RECORD DISPLAY
- DATABASE QUERY

CHART 13

In designing a student aid system, the processing related to data entry should be carefully defined. While the choice of edit processing only, or memo posting, or full transaction processing will usually be done in consonance with the practices of the computer center, data entry screens should follow the source documents from which the data are taken, edit messages should incorporate references to available information, and data elements needed for resolving edit problems should be displayed to the user. Similarly, record displays should be related to the uses of the data, and database queries should reflect the potential need for ad hoc reports or statistics.

One of the primary justifications for on-line processing is the resulting efficiencies of having clerical staff directly enter data rather than having an intermediate keypunch or key data entry step. Data entry efficiency depends on terminal response time and formats, and data entry effectiveness depends upon edit and data retrieval capabilities. One of the significant advantages of having data entry by skilled student aid staff is the ability to resolve data problems at the time of data entry rather than later. One of the disadvantages is the tendency not to have the data entry as well organized as it would be if the data were keypunched or keyed to magnetic media separately. In a well organized financial aid office, major clerical savings can occur for data entry and finding and displaying records. These savings can result in staff savings (often in the computer center for data entry), but more often are absorbed by increased workload or substituting professional counseling and review which were previosuly not done.

RELATIONSHIP OF STUDENT FINANCIAL AID SYSTEM TO OTHER INSTITUTIONAL SYSTEMS

INTER-SYSTEM COMMUNICATION

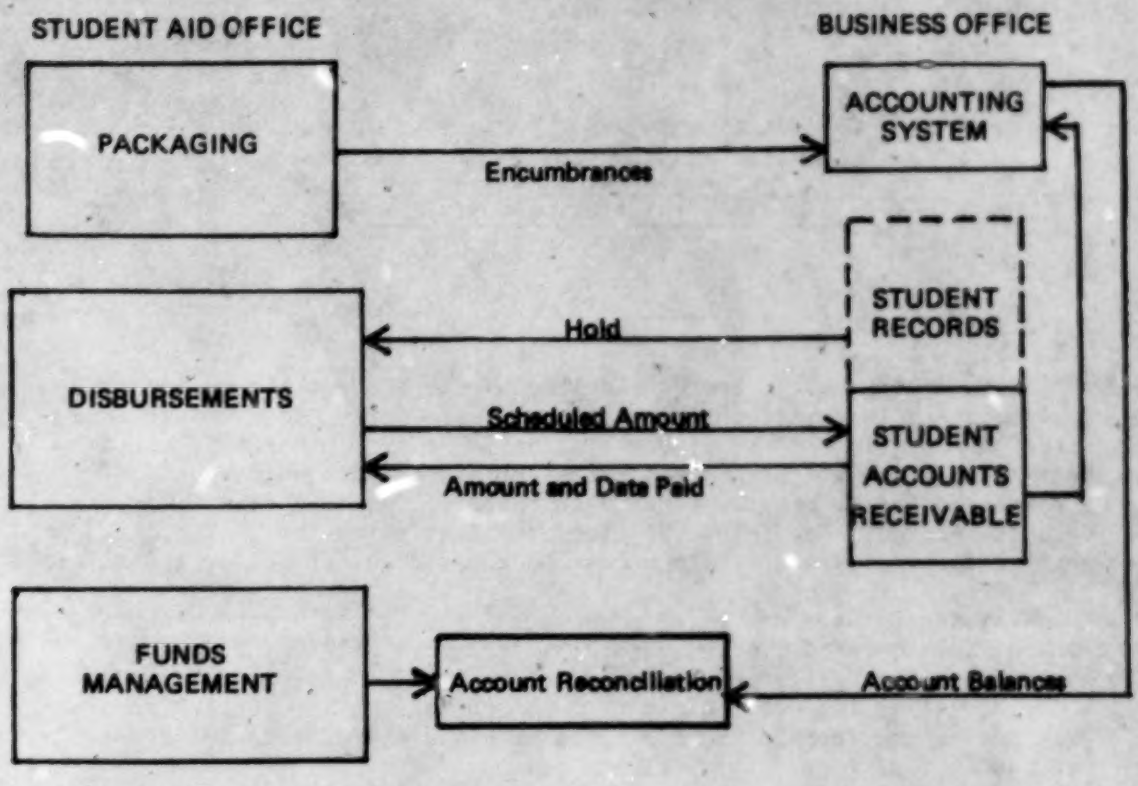
Some of the data elements required by the student aid system are available in other institutional systems. Similarly, some of the data needed by the student aid system are available from other application systems. These data are usually transferred between systems by an interface program, programs including capabilities to access files or database segments maintained by other application systems, or other applications systems directly accessing the student financial aid files.

Interfaces have one major advantage. They closely document the flow of data between systems, and permit responsibility for data entry and database integrity to be confined to the using office. Direct access to data, either through the use of a common database or special programming provisions, creates some processing efficiencies and avoids data redundancy, but increases system maintenance.

These issues can best be understood by some examples. The inter-office data flow between the business office and the student aid office is illustrated in Chart 14. As students are packaged, the business office makes encumbrances in the institutional accounting records for the student aid funds.

The business office may wish to hold the scheduling of payments or credits. In most system designs, this is done by a payment hold transaction. If the institution has a student accounts receivable system, then the disbursements function only schedules the payments and transfers the scheduled amount to the student accounts receivable system for credit to the student's account or payment to the student. This process is similar to processing vouchers. To assist the financial aid staff, particularly if on-line systems are used, the amount and date paid or credited should be made available to the staff. This flow of data represents the same data exchange which would occur in a manual system. The data flow can be accompanied by vouchers or journal entries so control totals can be verified by each office. The responsibility for processing is independent, well-defined, and functionally integrated with each office.

There is a tendency to include some business functions in the student aid system and some student aid functions in the business system. When this is done, the systems are no longer functionally integrated. That is, each system should represent the scope of responsibility of the "owning" office. The student accounts receivable function is a business office function and should not be included in the student aid system.



INTER-OFFICE DATA FLOW

CHART 14

Similarly, checks should not be written by the student aid office. Disbursements should be scheduled by the student aid office in accordance with federal and state regulations, but written by the business office.

The two systems should be reconciled by comparing account balances and student balances. The account balances in the accounting system should match or reconcile with the account balances in the student aid system. To facilitate this reconciliation, the student aid system should be able to provide fund balances by both the institutional account number and the program number. Because of differing regulations, procedures, and accounting periods, the program identification numbers for student aid processing will often be different from the institutional account numbers.

Although the two systems may use the same database, it is clear that each office should have a system which does not perform functions which are the responsibility of the other office. While data processors may cite reasons for "integrating" the applications, this integration of data should never violate the scope of responsibility of the affected offices.

The inter-system data flows are further diagrammed in Chart 15. In addition to the exchange of data with the business office systems, the student aid system need status information from the admissions systems, and demographic and enrollment data from student records. Student financial aid data are needed for student accounts receivable, the institutional accounting system and the institution research database. Data from national processors may be used by the student aid system, and data will be received from the federal BEOG Processing Center, state financial aid agencies and the Internal Revenue Service. Additional data may be received from state tax agencies, and federal and state student loan guarantors. Also, data may be required for lenders, the U.S. Office of Education, and state agencies, depending upon the types of student aid programs and institutional administrative policies. The system design should identify and accommodate these data flows. Unfortunately, the external requirements and the specifications for integration are being changed frequently. There is, however, a trend toward federal standards and to the exchange of machine-readable media such as magnetic tape and punched cards in lieu of lists and reports.

As the level of integration increases, the amount of maintenance generally increases. The student aid system is almost unique in the amount of data exchange with external agencies. The data required for exchange should be carefully examined to select a level of integration which is appropriate for the volume and type of processing which is done by the institution.

INTER-SYSTEM DATA FLOW

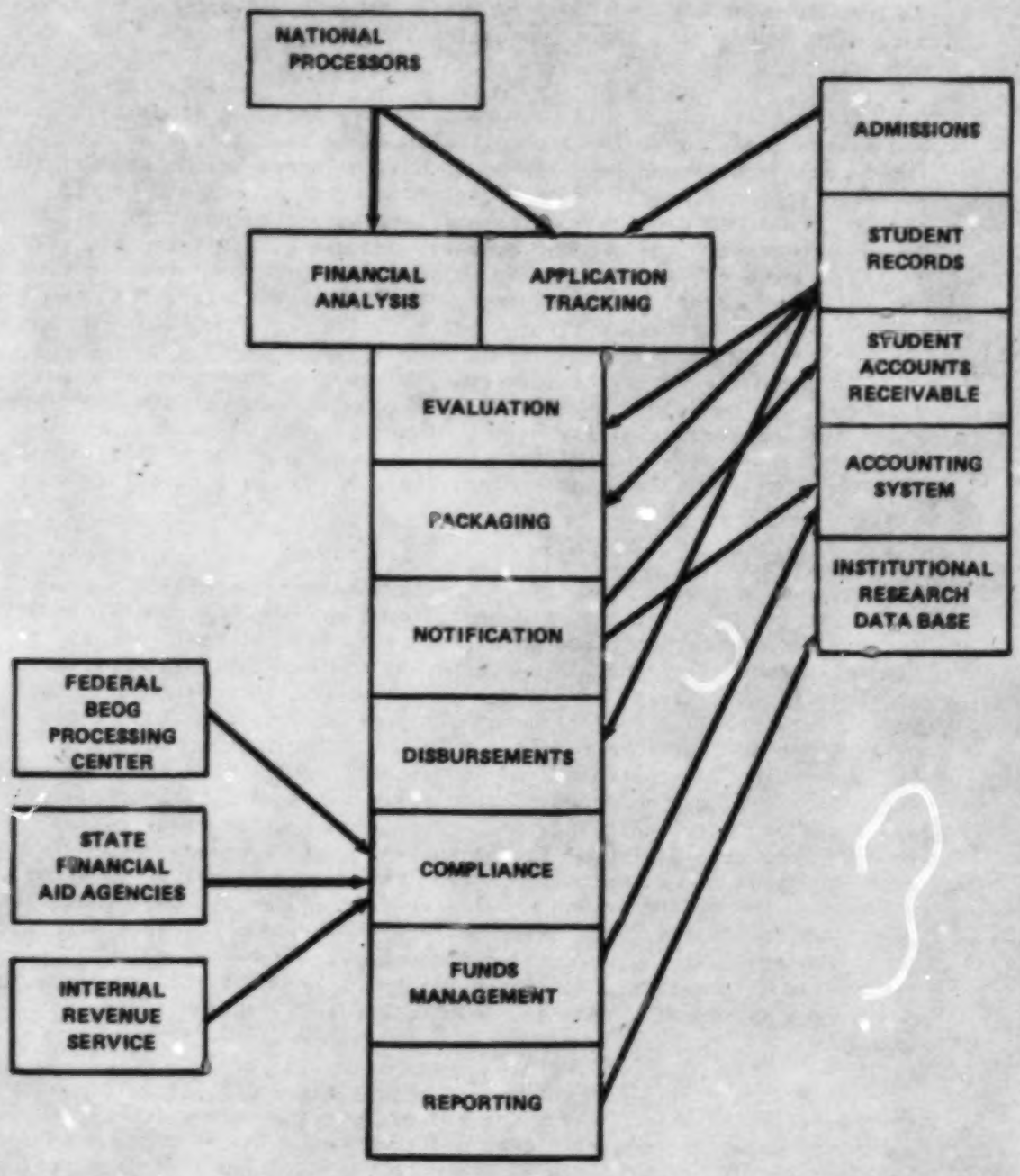


CHART 15

DATABASE INTEGRATION

Because changing the content of records frequently causes major and expensive programming changes to a system, data processors have developed several techniques to reduce the impact of these changes.

In conventional programming, the program contains a definition of the file, and records are retrieved from files using data access methods--specific manufacturer-supplied software which retrieves records from physical devices such as disk drives and tape drives and makes them available in main memory of the computer. The program file definition provides a name for each data element, its location in the record, and its format. To avoid duplicating this information for each program, higher level languages such as COBOL and PL/I have the capability of copying this description from a single file to facilitate programming. However, each time a data element is added, the programs would have to be recompiled, that is translated from the programming language to machine language. If the programmer has left blank space in the record, the data file itself does not have to be changed. If the programmer did not leave space, then the files will have to be changed to the new format. Sometimes this is a formidable task.

To prevent having to change the file description, several system designs include tables which have data element name, location, and representation. These tables are external to the program (see Chart 16). A single change to the table, with no changes to the programs themselves, permits data elements to be added to or deleted from the file. This is sometimes called a database management system by systems designers unfamiliar with current technology, but actually is table-driven applications software. By adding editing criteria, code tables, and column headings, this data definition table can be used with generalized data entry and file query systems, often called file management systems.

A database management system is part of the operating system--shared by all user programs, which stores data by its name or identification and has the ability to retrieve the data in several different sequences. The database can be changed by changing the tables. No programming changes may be required. The database supports all applications so data from any application may be made available to all applications. Thus admissions status is available for computers of several different manufacturers, enhancing transportability. Most application systems are either designed to the industry standard (IBM) or to the national standard (CODASYL).

A database management system prevents some system redundancy, reduces disk storage requirements, and facilitates program changes. Because it is common to all application systems, all data are potentially available to all applications. There are two disadvantages to these systems. First, in the past, database management systems have

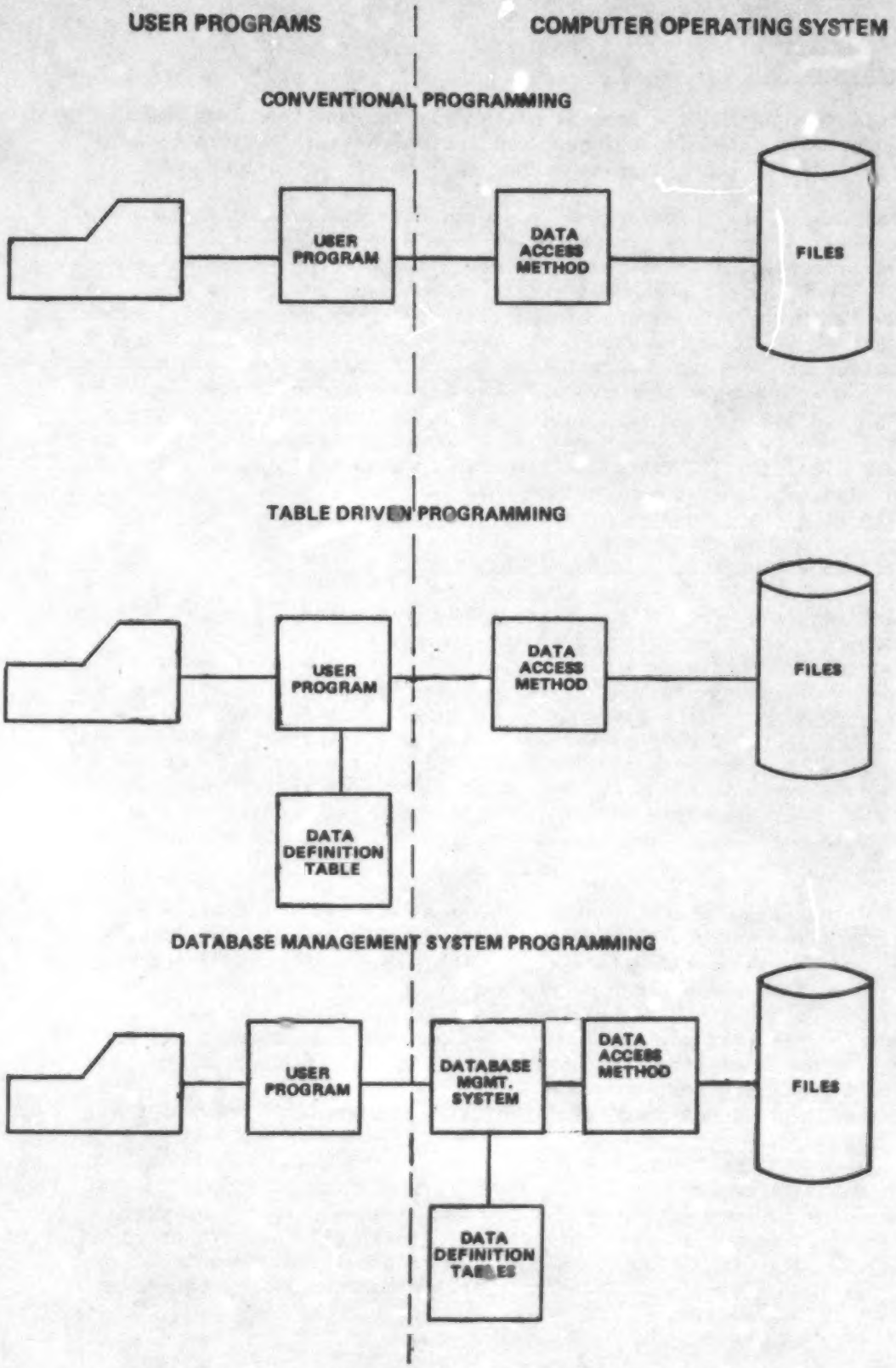

USER PROGRAMS
COMPUTER OPERATING SYSTEM
CONVENTIONAL PROGRAMMING
USER PROGRAM
DATA ACCESS METHOD
FILES
TABLE DRIVEN PROGRAMMING
USER PROGRAM
DATA ACCESS METHOD
FILES
DATA DEFINITION TABLE
DATABASE MANAGEMENT SYSTEM PROGRAMMING
USER PROGRAM
DATABASE MGMT. SYSTEM
DATA ACCESS METHOD
FILES
DATA DEFINITION TABLES

CHART 16

been inefficient and costly. Improved efficiency and lower computing costs have ameliorated this characteristic. Second, a higher level of proficiency has been required from the programmers. As more installations use these systems, more experienced programmers will become available to colleges and universities. Third, there has been no standard or system which has encompassed a majority of users. Until the federal government makes a major standardization effort or IBM develops a system acceptable to most users, there is some uncertainty about the future prospects of even the systems available today.

FILE STRUCTURE

ISSUE

Student aid system designers have used markedly different approaches to designing the file structure of a student aid system. The sharp differences in design have confused potential users, but again come from the complex nature of student financial aid processing. There are essentially two designs. One uses several different files, or segments of a database management system, which relates to the phases of student aid processing. Another uses a single record which contains all required data elements and is used throughout the processing cycle. The first approach has some data redundancy between files, but the second has unneeded data during much of the processing cycle. The first approach better supports statistical analysis, the second better supports student aid transcripts.

ALTERNATIVE APPROACHES

The need for data storage is illustrated in Chart 17. Essentially, the need for file storage spans three academic years. The application is processed during the first year, the award file is used for disbursement and compliance during the academic year, and history is used in the subsequent years. The chart relates the typical number of records which would be contained in the file during any period of time. This reflects a typical state university. Different types of institutions and different processing procedures could result in different volumes. Because the record sizes are significantly different for each of the phases, the number of characters typically required is shown with each file.

With a single record system, the record would be available during the entire period. By using the number of records, the record size, and time needed the total space need for data storage is illustrated in Chart 18. The most significant differentce in the two approaches is only after the academic year, when staged files permit small historical records to be retained in lieu of the larger single record. The difference can also be expressed in terms of cost. The average monthly data storage costs for both approaches are shown in Chart 19. These prices reflect typical college and university data center charges for minicomputers such as the DEC System 20, Prime PRIMOS, Harris Vulcan systems, or Hewlett-Packard Model 3000-III, and general purpose computers such as the IBM System 370, CDC Cyber, or Univac 1100 systems. (The recently announced IBM 4300 systems would have prices about one-third of those shown for general purpose computers.) One of the most important conclusions is that data storage charges are one of the least expensive operating costs of a student aid system, and cost reductions will continue as new hardware is introduced.

AVERAGE NUMBER OF RECORDS
10,000 STUDENTS STATE UNIVERSITY

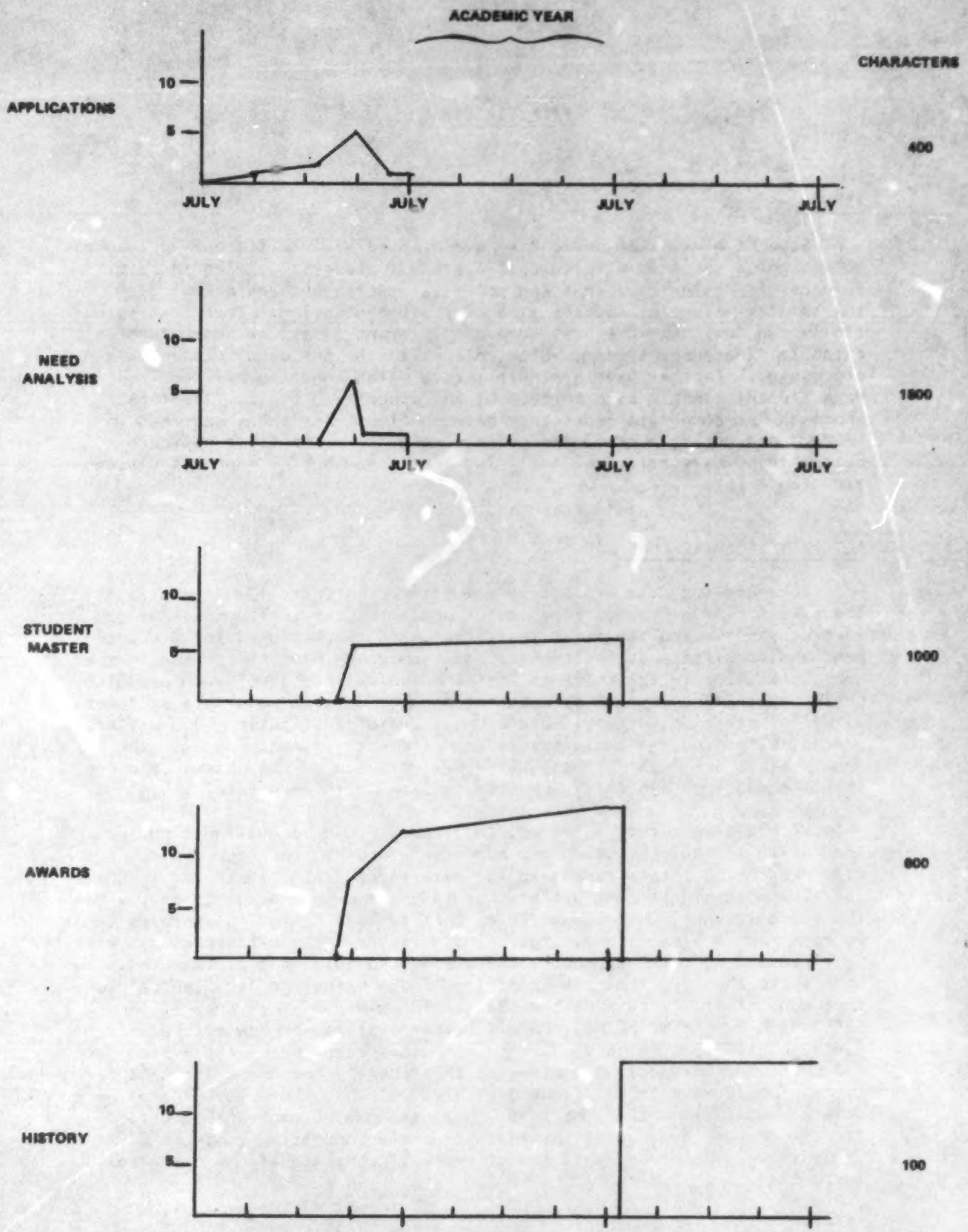
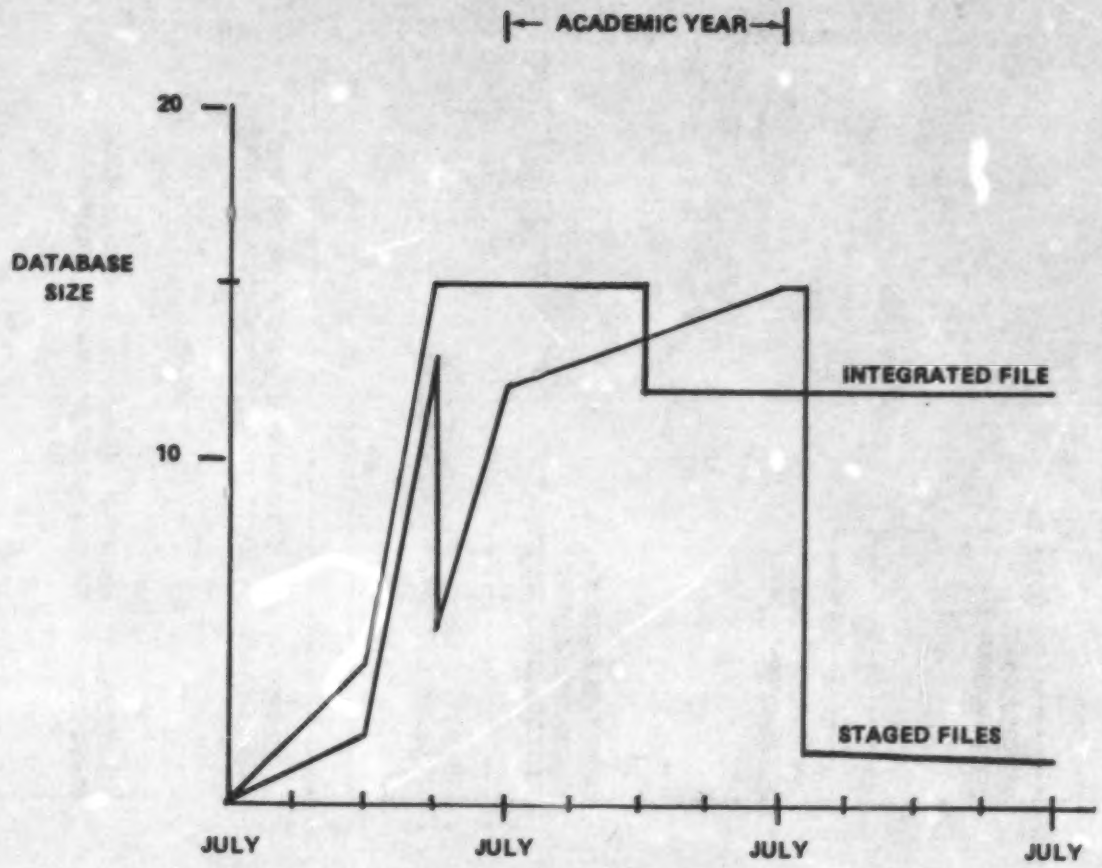


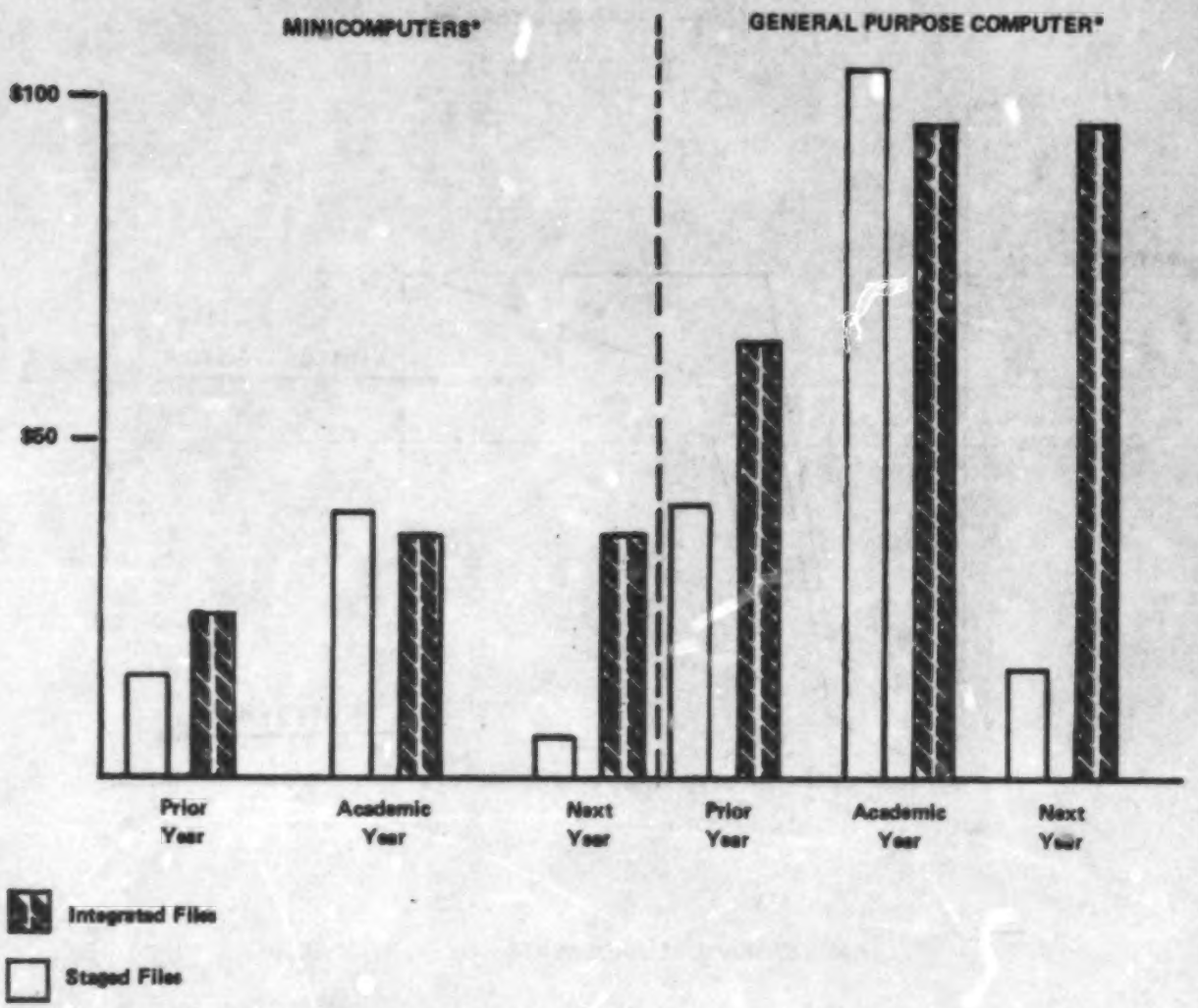
CHART 17



DATA STORAGE REQUIREMENTS 6000 AID RECIPIENTS

CHART 18

AVERAGE MONTHLY DATA STORAGE COSTS



*Minicomputer storage estimated on $3 per megabyte month. General purpose computer storage estimated as $8 per megabyte month.

CHART 19

While the file structure has little cost impact on the operating costs, the user should be aware some colleges and universities have limited disk storage, constraining a student aid file system implementation. Careful control of disk storage is as much a responsibility of the student aid office as it is the data center. With today's cost of disk storage devices, the costs of disk storage should not deter the student aid office from automating or using on-line approaches.

SUMMARY

In order to have a successful student aid system design, the student aid staff will have to identify carefully the functions and tasks which are needed, identify the processing mode appropriate to their practices, and determine an overall processing schedule acceptable to the institution. When these tasks are done, a useful systems requirements document can be written. More important, no vital function will have been omitted from the system.

In conjunction with data processors, the type of computer programs and the scope of the system can then be determined. This will identify a level of sophistication for the resulting system. Chart 20 shows some of the factors which affect the sophistication of the system. Sophistication itself is not an objective, but rather one measure of an application system. In general, sophisticated systems cost more to develop or purchase, cost more to maintain, perform more functions, produce a better quality output, and serve students, the institution, and external agencies better. The appropriate trade-off between sophistication, cost, and results must be made in light of institutional objectives. The student aid office must take responsibility for these decisions in conjunction with technical advice of the data center, financial analysis of the business office, and coordination with admissions, student records, academic offices, and auditors.

The criteria for design are summarized in Chart 21. The system should have all required major functional capabilities. The system should be flexible in order to adapt to changing policies and practices. The student aid system should first be integrated with the organization to support the accomplishment of institutional objectives, and secondly integrated with other data processing applications at the technical level. This technical integration is accomplished by exchanging data between applications--system integration--and creating, in the long range, a single database with appropriate assignments of responsibilities for data entry and data integrity.

The costs and cost savings include operational efficiency, the development and maintenance costs--which are and will be high until federal and state programs become stable--and last, and usually least important, computer efficiency. Only after performing the suggested analyses, identifying and making the trade-off decisions, and evaluating benefits and costs, can the student aid staff make their appropriate and needed contributions to the design or selection of a student aid system appropriate for accomplishing the institutional goal objectives. Hopefully, the framework provided by this guide will be helpful in this complex and arduous task.

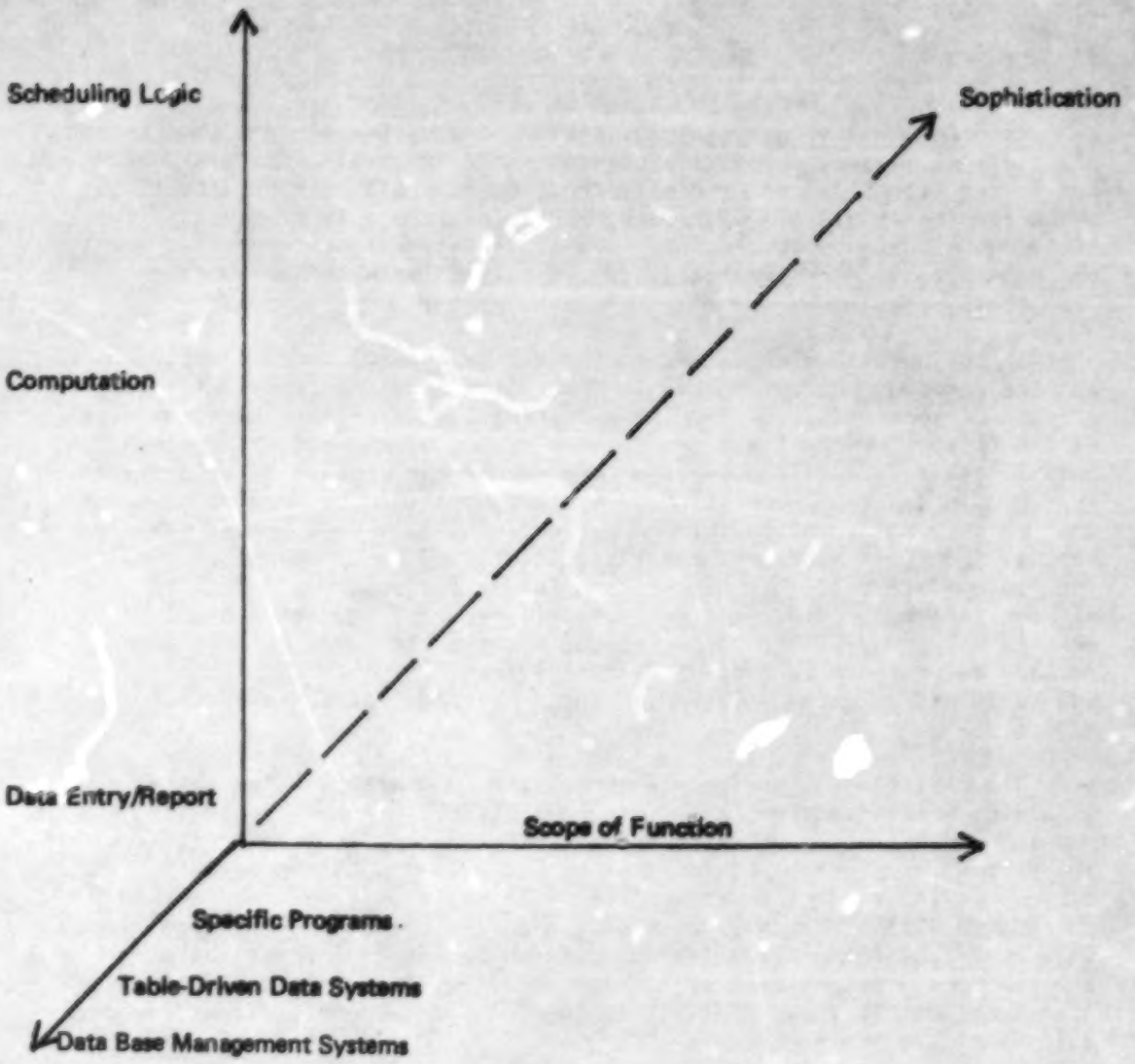

Scheduling Logic
Sophistication
Computation
Data Entry/Report
Scope of Function
Specific Programs
Table-Driven Data Systems
Data Base Management Systems

CHART 20

CRITERIA FOR DESIGN

BENEFITS

- FUNCTIONAL CAPABILITIES
- FLEXIBILITY
- ORGANIZATIONAL INTEGRATION
- SYSTEMS INTEGRATION
- DATABASE INTEGRATION

COSTS

- OPERATIONAL EFFICIENCY
- DEVELOPMENT AND MAINTENANCE COSTS
- COMPUTER EFFICIENCY

CHART 21